Applicant : Robert E. Kahn et al.   Attorney's Docket No.: 06154-008001
Serial No. : 08/720,092
Filed   · : September 27, 1996
Page   : 5

REMARKS

The applicant's remarks, below, are preceded by quotations of related comments of the examiner, in small, bold-face, type.

> **Claim 1 is rejected under 35 U.S.C. § 103 as being unpatentable over Antes, Gary M., "Let your 'knowbots' do the walking," Computerworld, May 13, 1991, pp(2), in view of Steinberg, Don, "Demon knowbots (intelligent software robots)," PC-Computing, v3, nl, pp(4), Jan, 1990.**
>
> **With respect to claim 1, The Examiner notes that "administrative Knowbots" that "police the system, keeping unauthorized users out" necessarily must intercede between system access requests (e.g., from other "Knowbots") and the underlying local system service facilities. Accordingly, the rejection of claim 1, as set forth in the last office action, is maintained. Additional new grounds of rejection are also set forth for claim 1, as detailed below.**
>
> **Antes discloses the invention substantially as claimed:**
>
> **Antes teaches a method for use in a distributed system for processing a mobile program that has the ability to move from node to node in the distributed system [e.g., page 1, line 24].**
>
> **Antes teaches an operating environment in each of the nodes that provides service facilities (e.g., databases) useful to the mobile program [e.g., page 1, line 30].**
>
> **However, Antes does not explicitly disclose the following additional limitations:**
>
> **Steinberg teaches an operating environment running a supervisor process [e.g., administrative knowbots, page 3, line 3] that allows the mobile program indirect access to make use of the service facilities [page 3].**
>
> **It would have been obvious to one of ordinary skill in the art at the time the invention was made to improve upon the system taught by Antes by implementing the improvements detailed above because it would provide Antes's system with the enhanced capability of keeping unauthorized users out [e.g., page 3, line 4].**

The applicant respectfully disagrees. Without conceding any of the examiner's points, the applicant notes that, in claim 1, the supervisor process "<u>allows</u> the mobile program <u>indirect access</u> to make use of the service facilities." (emphasis added) Conversely what is said in the Steinberg article is that the administrative Knowbots police the system by "keeping unauthorized users out." It would not have been obvious to a person of ordinary skill in the field to run a superviser process that allows <u>indirect access</u> by a mobile program to service facilities (as in claim 1) when the advice provided in the Steinberg article was to keep unauthorized users <u>out</u>, in

Applicant : Robert E. Kahn et al.
Serial No. : 08/720,092
Filed : September 27, 1996
Page : 6

Attorney's Docket No.: 06154-008001

other words, excluded directly. The claim recites a method that provides access in a certain way. Steinberg describes a system that prevents access. If anything, Steinberg's statement teaches away from the invention recited in claim 1.

> **Claims 1, 24 & 25 are rejected under 35 U.S.C. § 102(a) as being anticipated by Vanderburg, Glenn L., et al., "Tricks of the Java Programming Gurus," Sams.net Publishing, see Chapter 33, entire chapter, pp(14), 1996.**
>
> **As per claim 1:**
>
> **Vanderburg teaches a method for use in a distributed system for processing a mobile program that has the ability to move from node to node in the distributed system [e.g., see "Java" discussion beginning page 1]. The Examiner notes that mobile Java programs (i.e., applets) anticipate the claimed limitations.**
>
> **Vanderburg teaches an operating environment in each of the nodes that provides service facilities useful to the mobile program [e.g., a Java Virtual Machine; see also Java security and native method discussion beginning page 1],**
>
> **Vanderburg teaches an operating environment running a supervisor process [i.e., the Java Virtual Machine and associated security discussion, beginning page 1 | that allows the mobile program indirect access to make use of the service facilities [e.g., see "ExtensibleSecurityManager.java" code listing and associated native method discussion beginning page 5]. Java byte codes "indirectly interact" via the operating environment, as they are interpreted by the Java virtual machine at run time.**

The applicant has amended this application to claim priority from a co-pending United States patent application that was filed on May 30, 1995. Portions of the parent application that describe examples of what is recited in claim 1 have been copied into this application or were already part of this application, and other portions of the parent application were incorporated by reference. Thus, with respect to at least claims 1, 24, and 25, the applicant's effective filing date precedes the publication date of the Vandenburg article, and the rejection is moot.

> **As per claim 24:**
>
> **Vanderburg teaches a method for controlling interaction between a mobile program and an application running in an operating environment provided at a node of a distributed system comprising:**
>
> **defining a trusted portion of the operating environment which provides trusted services to the mobile program [e.g., see discussions of "Security in Native Method Libraries," page 1, and the "Extensible Security Manager," beginning page 4 - see code listing page 5-10],**

requiring portions of the application running in the operating environment to be registered as trusted, [e.g., see "registered" code as marked (underlined by Examiner) itl the code listing pages 7-10], and

permitting indirect interaction via the operating environment between the mobile program and the application running in the operating environment only if the portions of the application required to be registered have been registered [e.g., see "registered" code as marked (underlined by Examiner) in the code listing pages 7-10; e.g., see "Registering Specialized Security Managers" discussion beginning page 12]. The Examiner notes that the code listing is dated on pages 5 and 6 as Feb. 31, 1996. Java byte codes "indirectly interact" via the operating environment, as they are interpreted by the Java virtual machine at run time.

Please see the discussion above, which applies to claim 24.

As per claim 25:

This claim is rejected for the same reasons detailed above in the rejection of claim 24, and also for the following additional reasons:

Vanderburg teaches a method for enabling a mobile program to carry out defined functions including otherwise unsafe functions, though the use of extensions comprising:

coding safe extensions to an operating environment and to an interpretive language under which the mobile program runs, [e.g., see" ExtensibleSecurityManager.java" code listing beginning page 5, entire code listing and supporting discussion], and

permitting the mobile program to carry out the defined functions by making use of the extensions [e.g., see "registered" code as marked (underlined by Examiner) in the code listing pages 7-10; e.g., see "Registering Specialized Security Managers" discussion beginning page 12].

Please see the discussion above, which applies to claim 25.

In light of Applicant's arguments of record, the Examiner has reconsidered and withdrawn the remaining rejections. Dependent claim 2 appears to be allowable if rewritten to include all of the limitations of base claim 1, as the prior art does not appear to teach nor suggest the same or equivalent structure and function of the claimed "bastion object." This claimed software component is accorded the status of a "coined term" by the Examiner. Likewise, the claimed "connector objects" of claim 9 are examined in the context of a lexicon created by Applicant.

Accordingly, the scope of the aforementioned claimed coined terms is interpreted by the Examiner as being limited by the corresponding structure and function disclosed within the instant specification.

The applicant appreciates the indication that these claims would be allowable.

Applicant : Robert E. Kahn et al.
Serial No. : 08/720,092
Filed : September 27, 1996
Page : 8

Attorney's Docket No.: 06154-008001

Without conceding any of the examiner's points, the applicant has amended claims 9

through 12 to remove the word "connector" and to make clear that the claim encompasses not

only the connector as described in the specification, but any mechanism and objects that are

within the scope of the claim language, including equivalents. The applicant contends that the

claims are patentable as amended.

Claims 5-17 appear to be allowable, subject to the results of a final search.

The applicant acknowledges that claims 5-17 appear to be allowable.

Claims 18 & 19 are rejected under 35 U.S.C. § 103 as being unpatentable over Antes, Gary M., "Let your 'knowbots' do the walking," Computerworld, May 13, 1991, pp(2), in view of Steinberg, Don, "Demon knowbots (intelligent software robots)," PC-Computing, v3, n1, pp(4), Jan, 1990, and further in view of Orfali et al., "Client/Server Programming with CORBA Objects," OS/2 Magazine, Sept. 1994, pp(8).

As per independent claim 18:

Antes, as modified by Steinberg, teaches the invention substantially as claimed.

Antes, as modified by Steinberg, teaches a method for aiding communication with a mobile program executing in operating environments provided at nodes of a distributed system (as discussed above in the rejection of claim 1).

However, Antes & Steinberg do not explicitly disclose the following additional limitations:

Orfali teaches maintaining a name space that uniquely identifies types of information to be interchanged as part of the communication [e.g., page 3, #7, i.e., "Register the run-time objects with the implementation repository" - see the disclosed "object reference"], and using a name within the name space to identify a type of information to be interchanged [e.g., page 3, #7, i.e., "Register the run-time objects with the implementation repository"].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to improve upon the combined system taught by Antes & Steinberg by implementing the improvements detailed above because it would provide their system with the enhanced capability of knowing which object classes are supported on a particular server [Orfali, page 3, discussion #7].

As per claim 19:
Antes, as modified by Steinberg and Orfali, teaches the mobile program registers an interface which includes the name of a type of information that is to be interchanged [e.g., Orfali, page 3, #7, i.e., "Register the run-time objects with the implementation repository"].

The applicant discussed Antes and Steinberg in its prior response. With respect to Orfali,

the applicant's claim 18 requires "using a name space that uniquely identifies types of

Applicant : Robert E. Kahn et al.
Serial No. : 08/720,092
Filed · : September 27, 1996
Page : 9

Attorney's Docket No.: 06154-008001

information to be interchanged." The applicant has been unable to locate a copy of the CORBA article by Orfali cited by the examiner and requests the examiner to provide a copy for the applicant. However, the applicant is generally familiar with Orfali's work on client-server models for object-oriented programming and believes that the naming described in his work does not apply to communication with mobile programs as recited in claims 18 and 19.

The dependent claims are patentable for the same reasons as the independent claims on which they depend.

Applicant asks that all claims be allowed. Please apply any excess charges or credits to Deposit Account No. 06-1050, reference 06154-008001.
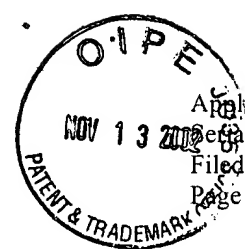
Respectfully submitted,

Date: _____

David L. Feigenbaum
Reg. No. 30,378

Fish & Richardson P.C.
225 Franklin Street
Boston, Massachusetts 02110-2804
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

20524556.doc

Applicant : Robert E. Kahn et al.
Serial No. : 08/720,092
Filed : September 27, 1996
Page : 10

Attorney's Docket No.: 06154-008001

Version marked to show changes

In the specification

On page 1, ahead of line 5, insert the following:

--This application is a continuation-in-part of United States Patent Application Serial

Number 08/453,486, filed May 30, 1995, and incorporated by reference.--

Insert the following material that appeared in the parent case and was incorporated by

reference in this application:

On page 10, line 4, at the end of the line, insert:

--A participant at a computer (e.g., a person or a process or a Knowbot program running

on the computer) may define the task to be executed. Information related to the defined task may

be embedded in one of the Knowbot programs. Task information contained in Knowbot

programs may be interpreted. The step of advancing the execution of tasks may include:

creating additional Knowbot programs, interacting with other Knowbot programs, interacting

with repositories of digital objects; querying a database of information; applying stored expert

knowledge; protocol transformation; providing a directory service identifying other locations

which provide services related to the task; making a security determination; providing a "what's

new" service to identify newly available information; providing a "clipping" service which

extracts information; providing a version control service for managing versions of the Knowbot

program; responding to actions of a user by obtaining execution of the task at another location

without necessarily indicating to the user that the execution occurred at the other location;

providing a method for determining the structure of a Knowbot program (e.g., to determine if a

digital object is contained in the Knowbot program and to access the element; creating a

derivative work from an existing work; generating a Knowbot program that contains another

Applicant : Robert E. Kahn et al.
Serial No. : 08/720,092
Filed · : September 27, 1996
Page : 11

Attorney's Docket No.: 06154-008001

Knowbot program, or a digital object, or other data; and passing a message from a source Knowbot program to a target Knowbot program at a different location.

The information concerning a task to be done may include interpretable or executable instructions. The Knowbot program may include data in the form of a digital object or a Knowbot program (a Knowbot program may itself be a digital object). The digital object may include protocol transformation information.--

On page 30, ahead of line 3, insert the following:

--Knowbot programs and Knowbot service stations within the Knowbot service environment may be designated as "qualified entities," that provide certain assurances on system behavior although such assurances may not be possible for all service stations. Various levels of assurance may also be designated. For a service station, qualification amounts to registering the existence of the service station within the Knowbot service environment. The party applying for such registration would agree not to modify (or otherwise tamper with) the service station except as authorized. The service station could contain mechanisms to determine its own integrity (e.g., if it has been altered or its operation otherwise modified).

Knowbot programs arriving at a qualified service station may be allowed to be handled there without regard for the qualification of the service station which originated the program, provided the cost of execution is acceptable. For executions that are costly, electronic payment may be required to be authorized or included. If digital objects subject to rights (e.g., computer programs) or relating to rights (e.g., contracts or deeds) are requested for access, the Knowbot program may only be allowed to be returned to qualified service stations at a certain level of trust. A test may be made prior to performing the access to determine if the source service station of the program is qualified or not. Or, in a less restrictive scheme, the program may simply be limited in what can be done with the digital object.

Applicant : Robert E. Kahn et al.   Attorney's Docket No.: 06154-008001
Serial No. : 08/720,092
Filed   : September 27, 1996
Page   : 12

For Knowbot programs, qualification may mean simply that a qualified service station was the creator of the program, assigned it a globally unique identifier, and maintains a copy of the Knowbot program along with any associated information. When a Knowbot program is deleted, a status message is returned to the service station that created it. This message may be deferred in delivery if that station is not available. Other mechanisms for notification in the event of long-lived Knowbot programs and relatively short-lived stations involve transfer of responsibility by notification to other stations.--

On page 45, line 1, ahead of the heading, insert the following:

--A Knowbot program could be used to protect sensitive or confidential information. In this case, the terms and conditions might be based on the identity of the user and his need to know. Knowbot service stations could include third-party value-added service providers that facilitate the transfer of information from information providers to information consumers.--

In the Claims

9. (twice amended) A method for enabling communication with a mobile program running in a distributed system, a mobile program service station, an extension, or another application, comprising

providing a [connector] mechanism which permits each of mobile program, the mobile program service station, the extension, and the other application to identify services that it provides, and permits each of them to find services that it needs, and

enabling the mobile program to communicate with mobile program service stations via [connector] objects associated with the [connector] mechanism.

10. (twice amended) The method of claim 9 in which each of the [connector] objects is provided by a supervisor process running in the distributed system and prevents uncontrolled access to a needed service.

11. (amended) The method of claim 9 in which the [connector] mechanism includes a [connector] broker and a [connector] manager.

12. (amended) The method of claim 9 in which the [connector] objects are data typed.

--26. (new) The method of claim 9 in which the mechanism comprises a connector mechanism, and the objects comprise connector objects.--